Appendix A. IPv6 Transition Assessment Guide

TRANSITION GUIDE PREAMBLE

The purpose of this Appendix is to help Program Managers evaluate the affects of IPv6 transition on their programs. Education on IPv6 needs to occur to enhance understanding of the implications associated with network layer transition. While this paper is not a detailed technical reference for IPv6 implementation it does promote a broad understanding of the transition challenges associated with building IPv6 capability into current systems. The paper is intended to spur thinking about and planning for changes that will result from IPv6 adoption as well as create awareness of new capabilities afforded by this protocol.

The template in Appendix B provides a checklist for evaluating how a program will transition to IPv6. It should be used in conjunction with this white paper to highlight areas requiring attention. The template will also provide important programmatic information to enterprise transition planners. Completed templates will be aggregated and used to organize transition efforts and validate requirements for additional funding.

Careful review and planning cannot capture all IPv6 implementation issues or guarantee system performance. The only way to be truly certain that a system or application is IPv6 capable is to test it on an IPv6 network. An IPv6 test network exists on the Defense Research and Engineering Network (DREN). Marine Corps agencies supporting IPv6 testing through the DREN are Marine Corps Network Operations Security Command (MCNOSC), Marine Corps Warfighting Lab (MCWL), and Marine Corps Tactical Systems Support Activity (MCTSSA). Program Managers are encouraged to seek support from these activities to test and evaluate the IPv6 capability of their hardware and software.

1 NETWORK COMMUNICATIONS

1.1 Seven Layer Model

Computer communications requires

(1) A connection,
(2) Protocols establishing rules for transferring information, and
(3) Application services that provide an interface with users or applications.

The Open Systems Interconnection (OSI) Reference Model describes how two points communicate on a network. Conceptually, each endpoint must complete the seven steps of the OSI Reference model for successful communications to take place. Figure 6[2] illustrates the OSI Reference Model and shows standard services, protocols and physical medium that are commonly associated with each of the seven layers. The lower three layers of the model are used when a message travels between two points on a network; the upper four levels are used at the source and destination to complete the information exchange. In addition to enabling features described in Chapter 3 of this Appendix, Internet Protocol (version 6 or version 4) provides the Network Layer service that contributes to computer communications by establishing rules for transferring information. Although "Internet Protocol" refers to a specific OSI Layer 3 entity, the Internet Protocol Suite commonly referred to as "TCP/IP" consists of both Layer 3 and Layer 4 protocols, of which Transmission Control Protocol (TCP) and IP are the most prevalent.

The transition to IPv6 will be quite similar to the Y2K transition. All operating systems, middleware, and applications will need to be examined for compatibility with the new standard and with published transition mechanisms. This transition is not a software-only event: many routers, switches, and other network devices will require replacement.

[2] Figure 6 from SearchNetworking.com website URL: http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci523729,00.html. March 2004.

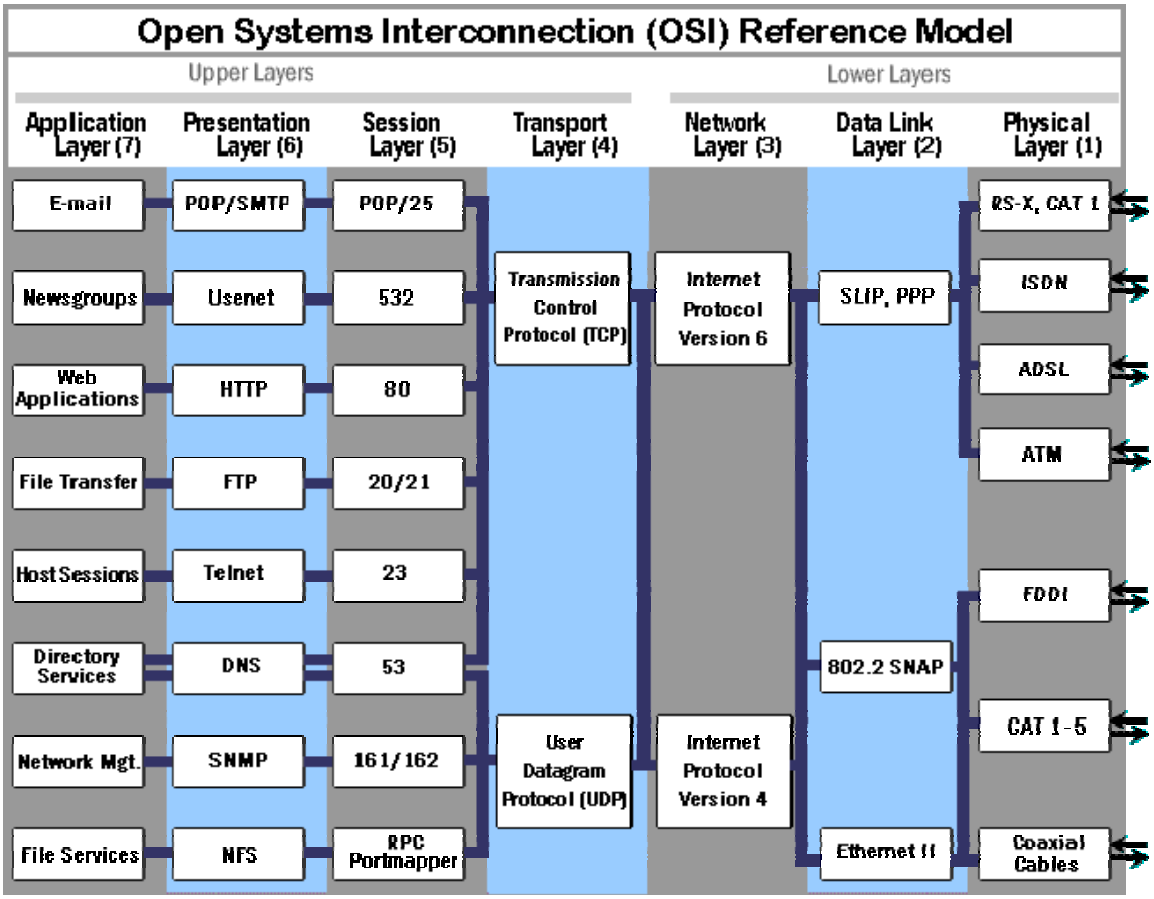

Figure 6. OSI Seven Layer Model

1.1.1 Internet Protocol

Understanding the role of IP in networking is critical to assessing the impact of transition from Internet Protocol version 4 (IPv4) to Internet Protocol version 6 (IPv6). IP is used at every connection point, or interface, to the network. Interfaces are assigned addresses based on the Internet protocol used and determine if traffic on the network pertains to them by examining the destination IP address of all traffic received. IP is integral to the function of networking appliances such as routers, switches, and bridges. Any application that sends information over a network must pass through an interface to do so; how that application accesses the interface determines whether changing from IPv4 to IPv6 will affect its operation.

Internet Protocol (IP) provides a common logical interface to different networks. This logical interface takes the form of an IP header appended to a manageable portion of the data that must be transported. This combination of header and data is referred to as a packet. Once a packet successfully negotiates the network and reaches its intended destination the information contained in the packet is used according to the rules established by the higher-level protocols involved in the exchange (see Layers 4 through 7 of Figure 6).

IP provides best effort delivery of packets of information across a network. The Internet Protocol itself does not ensure these packets arrive in correct order or that they

even arrive at all. Other protocols are used to guarantee delivery and reconstruct information by correctly combining the IP packets at their destination.

1.1.2 How Internet Protocol is used

Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) packets are created by appending a header (described in section 1.2) to a "payload" containing a manageable portion of the data being transported. Header fields carry information about the format of data contained in the payload and how the packet should be routed. Headers are examined at each node in the network that needs to determine routing. The payload contains the information that is being transported; the information in the payload is formatted using an upper layer protocol, such as Transfer Control Protocol (TCP) or User Datagram Protocol (UDP). In the case of IPv6, the payload can also contain extension headers. Extension headers are optional and carry additional instructions for processing the data contained in the payload; they offer the distinct advantage of being transparent to nodes in the path to the destination, facilitating efficient routing through the network.

To traverse portions of the network each IP Packet is encapsulated in additional Link Layer information needed to negotiate the paths between each node. The Link Layer Protocol used determines the maximum packet size allowed. For instance, Ethernet frames longer than 1518 bytes in length are ignored by Ethernet network interfaces. IP prevents oversize packets by fragmenting the data to pieces that will fit the maximum Message Transmission Unit (MTU) of the Link Layer Protocol. Figure 7 illustrates encapsulation of an IPv6 packet into a Link Layer Frame.

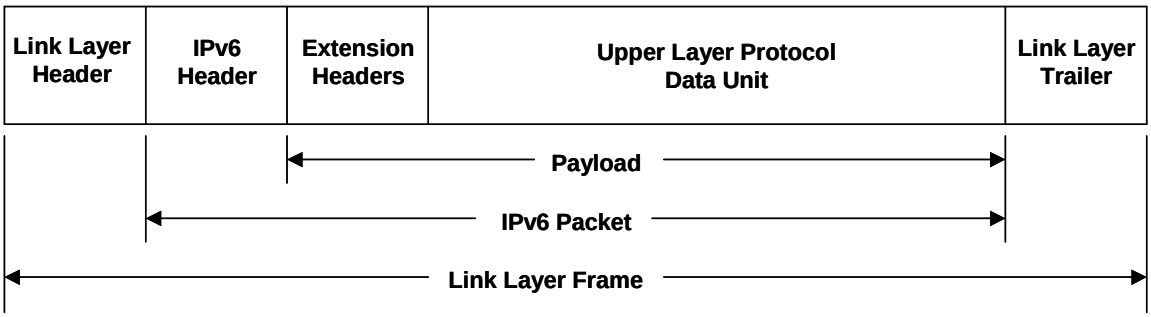

Figure 7. IPv6 Packet Encapsulation

Current IPv4 routers may be configured to support the recommended minimum MTU of 576 octets. For IPv6 routers the minimum length allowed is 1280 octets and the recommended MTU is 1500 octets; this larger MTU will allow Ethernet frames to carry forward without fragmentation.[3] The effects of increased packet size need to be evaluated on constrained bandwidth connections to determine the optimum MTU.

1.2 IP Headers

[3] Data Networks, IP and the Internet [electronic resource]: Protocols, Design and Operation by Martin P. Clark. Chichester, England; Hoboken, NJ: Wiley, 2003. Page 199.

IPv4 headers and IPv6 headers are not interoperable. IPv6 is not a superset of functionality that is backward compatible with IPv4. A host or router must use an implementation of both IPv4 and IPv6 in order to recognize and process both header formats.[4]

IPv6 headers are designed to optimize the use of header space and minimize the processing time needed at intermediate nodes in the transmission path. Unlike IPv4 headers, IPv6 headers are fixed in length. Address length increased from 32 bits with IPv4 to 128 bits with IPv6, a four-fold increase in bytes used for addressing. However, IPv6 headers are not significantly larger than IPv4 headers; overall IPv6 header length remains fixed at 40 bytes (or octets) while IPv4 headers can vary in length from 20 to 60 bytes. Table 8 identifies some of the key differences between header implementations for each version.

IPv4	IPv6
Source and destination addresses are 32 bits (4 bytes or octets) in length.	Source and destination addresses are 128 bits (16 bytes or octets) in length.
Header length varies from 20 to 60 bytes	Header length is fixed at 40 bytes
IPSec support is optional.	IPSec support is required.
No identification of packet flow for QoS handling by routers is present within the IPv4 header.	Packet flow identification for QoS handling by routers is included in the IPv6 header using the Flow Label field.
Both routers and the sending host do fragmentation.	Fragmentation is only done by the sending host, not by intermediate routers.
Header includes a checksum.	Header does not include a checksum.
Header includes options.	All optional data is moved to IPv6 extension headers.
Must be configured either manually or through DHCP.	Does not require manual configuration or DHCP.
Routing architecture must support a 576-byte packet size (possibly fragmented).	Routing architecture must support a 1280-byte packet size (without fragmentation).

Table 8. IPv4 and IPv6 Header Comparison

1.2.1 IPv4 Header

IPv4 headers are structured as shown in Figure 8. The highlighted portions are removed from IPv6 headers.

[4] Microsoft Windows Server 2003 White Paper, September 2003. Page 2.

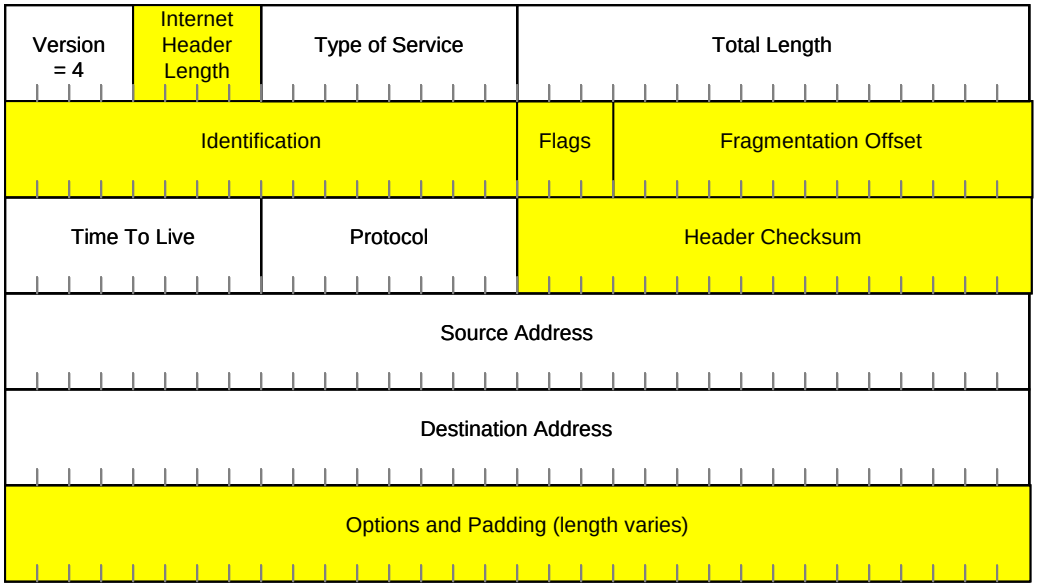

Figure 8. IPv4 Header

Version – Indicates the version of IP and is set to 4. The size of this field is 4 bits.

Internet Header Length – Indicates the number of 4-byte blocks in the IPv4 header. The size of this field is 4 bits. Because an IPv4 header is a minimum of 20 bytes in size, the smallest value of the Internet Header Length (IHL) field is 5 (5x4). IPv4 options can extend the minimum IPv4 header size in increments of 4 bytes. The maximum size of the IPv4 header including options is 60 bytes (15×4).

Type of Service – Indicates the desired service expected by this packet for delivery through routers across the IPv4 network. The size of this field is 8 bits, which contain bits for precedence, delay, throughput, and reliability characteristics.

Total Length – Indicates the total length of the IPv4 packet (IPv4 header + IPv4 payload) and does not include link layer framing. The size of this field is 16 bits, which can indicate an IPv4 packet that is up to 65,535 bytes long.

Identification – Identifies this specific IPv4 packet. The size of this field is 16 bits. The Identification field is selected by the originating source of the IPv4 packet. If the IPv4 packet is fragmented, all of the fragments retain the Identification field value so that the destination node can group the fragments for reassembly.

Flags – Identifies flags for the fragmentation process. The size of this field is 3 bits, however, only 2 bits are defined for current use. There are two flags – one to indicate whether the IPv4 packet might be fragmented and another to indicate whether more fragments follow the current fragment.

Fragment Offset – Indicates the position of the fragment relative to the original IPv4 payload. The size of this field is 13 bits.

Time to Live – Indicate the maximum number of links on which an IPv4 packet can travel before being discarded. The size of this field is 8 bits. The Time-to-Live field (TTL) was originally used as a time count with which an IPv4 router determined the length of time required (in seconds) to forward the IPv4 packet, decrementing the TTL accordingly. Modern routers almost always forward an IPv4 packet in less than a second and are required by RFC 791 to decrement the TTL by at least one. Therefore, the TTL becomes a maximum link count with the value set by the sending node. When the TTL equals 0, an ICMP Time Expired message is sent to the source IPv4 address and the packet is discarded.

Protocol – Identifies the upper layer protocol. The size of this field is 8 bits. For example, TCP uses a Protocol of 6, UDP uses a Protocol of 17, and ICMP uses a Protocol of 1. The Protocol field is used to demultiplex an IPv4 packet to the upper layer protocol.

Header Checksum – Provides a checksum on the IPv4 header only. The size of this field is 16 bits. The IPv4 payload is not included in the checksum calculation as the IPv4 payload and usually contains its own checksum. Each IPv4 node that receives IPv4 packets verifies the IPv4 header checksum and silently discards the IPv4 packet if checksum verification fails. When a router forwards an IPv4 packet, it must decrement the TTL. Therefore, the Header Checksum is recomputed at each hop between source and destination.

Source Address – Stores the IPv4 address of the originating host. The size of this field is 32 bits.

Destination Address – Stores the IPv4 address of the destination host. The size of this field is 32 bits.

Options – Stores one or more IPv4 options. The size of this field is a multiple of 32 bits. If the IPv4 option or options do not use all 32 bits, padding options must be added so that the IPv4 header is an integral number of 4-byte blocks that can be indicated by the Internet Header Length field.

1.2.2 IPv6 Header

Figure 9 shows the fields in the IPv6 Header. The number of fields has been reduced and the address space has quadrupled. Overall header length is now fixed at 40 octets, thus eliminating the need for the “Internet Header Length” field used in IPv4 headers. Other changes are discussed in the description of each header field below.

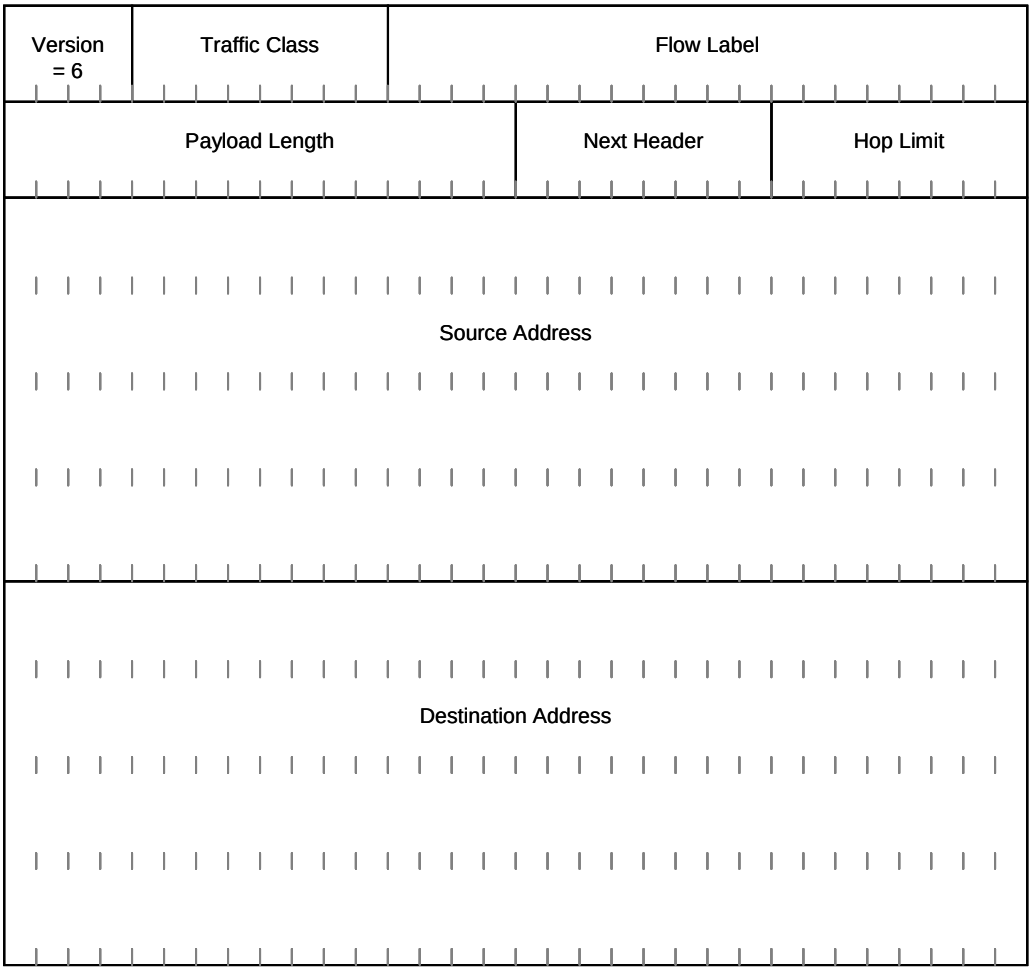

Figure 9. IPv6 Header

Version – 4 bits are used to indicate the version of IP and is set to 6.

Traffic Class – Indicates the class or priority of the IPv6 packet. The size of this field is 8 bits. The Traffic Class field provides similar functionality to the IPv4 Type of Service field. In RFC 2460, the values of the Traffic Class field are not defined. However, an IPv6 implementation is required to provide a means for an application layer protocol to specify the value of the Traffic Class field for experimentation.

Flow Label – Indicates that this packet belongs to a specific sequence of packets between a source and destination, requiring special handling by intermediate IPv6 routers. The size of this field is 20 bits. The Flow Label is used for non-default quality of service connections, such as those needed by real-time data (voice and video).

Payload Length – Indicates the length of the IPv6 payload. The size of this field is 16 bits. The Payload Length field includes the extension headers and the upper layer Protocol Data Unit (PDU).

Next Header – Indicates either the first extension header (if present) or the protocol in the upper layer PDU (such as TCP, UDP, or ICMPv6). The size of this field is 8 bits. When indicating an upper layer protocol above the Internet layer, the same values used in the IPv4 Protocol field are used here.

Hop Limit – Indicates the maximum number of links over which the IPv6 packet can travel before being discarded. The size of this field is 8 bits. The Hop Limit is similar to the IPv4 TTL field except that there is no historical relation to the amount of time (in seconds) that the packet is queued at the router. When the Hop Limit equals 0, an ICMPv6 Time Exceeded message is sent to the source address and the packet is discarded.

Source Address –Stores the IPv6 address of the originating host. The size of this field is 128 bits.

Destination Address – Stores the IPv6 address of the current destination host. The size of this field is 128 bits. In most cases the Destination Address is set to the final destination address. However, if a Routing extension header is present, the Destination Address might be set to the next router interface in the source route list.

2 HOW THE NETWORK LAYER IS USED

2.1 Assigning IP Addresses

Any device or software application that intends to communicate across the network must have an IP address to originate from and an IP address or range of addresses to reach. Devices themselves do not have addresses; hardware interfaces and software ports have addresses. IP addresses can be statically assigned by administrators or dynamically assigned through Dynamic Host Configuration Protocol (DHCP) services. IPv4 and IPv6 each have implementations of DHCP. IPv6 further supports Stateless Address Configuration (see section 3.3.4).

2.2 Application Program Interfaces

Communication between a client program and a server program in a network is accomplished through the use of sockets. A socket is defined as "the endpoint in a connection." Sockets are created and used with a set of programming requests or "function calls" sometimes called the sockets application programming interface (API). Sockets can also be used for communication between processes within the same computer. Sockets can be implemented as “connectionless” or “connection-oriented.” Connectionless sockets use datagrams and exist only long enough for a single exchange to take place while connection-oriented sockets use streams and remain aware until terminated. The address of a socket in the Internet domain consists of the IP address of the host machine and a port number on that host. Port numbers are 16 bit unsigned integers. Standard services have established ports so that clients will know their addresses. For example, the port number for the FTP server is 21.

The formal requests for services and means of communicating with other programs that a programmer uses in writing an application program is called the application program interface. An API is the specific method prescribed by a computer operating system (OS) or by an application program for making requests of the operating system or other applications. Applications can be developed to use the services provided by operating systems to establish communications links with other hosts or applications. Alternately, some applications do not use the standard function calls in the operating system. Instead, connections to other hosts and applications may be established using hard-coded protocol implementations, ports, and addresses. Identifying these non-standardized applications is important since upgrading to operating systems with IPv6 capability will not alter the procedures coded in programs. Application programs with hard-coded API implementations will need to be replaced or undergo additional development, testing and certification. Because this is not readily apparent in an application, careful analysis of source code is required in order to make this determination. A very common hard-coded implementation is the loopback address, 127.0.0.0. Hardcoded addresses and implementations are among the principal reasons that IPv6 transition is an involved process.

3 IPV6 IMPLEMENTATION

3.1 Addressing

IPv4 addresses are represented in dotted-decimal format. This 32-bit address is divided along 8-bit boundaries. Each set of 8 bits is converted to its decimal equivalent and separated by periods. For IPv6, the 128-bit address is divided along 16-bit boundaries, and each 16-bit block is converted to a 4-digit hexadecimal number and separated by colons. The resulting representation is called colon-hexadecimal. IPv6 representation can be further simplified by removing the leading zeros within each 16-bit block. However, each block must have at least a single digit.

Some types of addresses contain long sequences of zeros. To further simplify the representation of IPv6 addresses, a contiguous sequence of 16-bit blocks set to 0 in the colon hexadecimal format can be compressed to "::", known as *double-colon*. For example, the link-local address of FE80:0:0:0:2AA:FF:FE9A:4CA2 can be compressed to FE80::2AA:FF:FE9A:4CA2.

The prefix is the part of the address that indicates the bits that have fixed values or are the bits of the network identifier. Prefixes for IPv6 subnet identifiers, routes, and address ranges are expressed in the same way as Classless Inter-Domain Routing (CIDR) notation for IPv4. An IPv6 prefix is written in *address*/*prefix-length* notation. For example, 21DA:D3::/48 is a route prefix and 21DA:D3:0:2F3B::/64 is a subnet prefix. [5]

There are three types of IPv6 addresses – Unicast, Multicast, and Anycast. Table 9 compares addresses used in IPv4 to those used in IPv6.

IPv4 Address	IPv6 Address
Internet address classes	Not applicable in IPv6
Multicast addresses (224.0.0.0/4)	IPv6 multicast addresses (FF00::/8)
Broadcast addresses	Not applicable in IPv6. "Link-scope all-hosts multicast" address, FF02::1 corresponds to IPv4 subnet-local address, 255.255.255.255
Not applicable in IPv4	Anycast addresses
Unspecified address is 0.0.0.0	Unspecified address is ::
Loopback address is 127.0.0.1	Loopback address is ::1
Public IP addresses	Global unicast addresses
Private IP addresses (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16)	Site-local addresses (FEC0::/10)
Autoconfigured addresses (169.254.0.0/16)	Link-local addresses (FE80::/64)
Text representation: Dotted decimal notation	Text representation: Colon hexadecimal format with suppression of leading zeros and zero compression. IPv4-compatible addresses are expressed in dotted decimal notation.

[5] Microsoft Windows Server 2003 White Paper, September 2003. Pages 8-10.

IPv4 Address	IPv6 Address
Network bits representation: Subnet mask in dotted decimal notation or prefix length	Network bits representation: Prefix length notation only
DNS name resolution: IPv4 host address (A) resource record	DNS name resolution: IPv6 host address (AAAA) resource record
DNS reverse resolution: IN-ADDR.ARPA domain	DNS reverse resolution: IP6.ARPA domain

Table 9. IPv4 and IPv6 Address Convention Comparison[6]

3.1.1 Unicast Addresses

A unicast address identifies a single interface within the scope of the type of unicast address. With the appropriate unicast routing topology, packets addressed to a unicast address are delivered to a single interface.

The following types of addresses are unicast IPv6 addresses:

Global Unicast Addresses. Global unicast addresses are equivalent to public IPv4 addresses. They are globally routable and reachable on the IPv6 portion of the Internet. Unlike the current IPv4-based Internet, which is a mixture of both flat and hierarchical routing, the IPv6-based Internet has been designed from its foundation to support efficient, hierarchical addressing and routing. The scope, the region of the IPv6 internetwork over which the address is unique, of a global unicast address is the entire IPv6 Internet.

Local-Use Unicast Addresses. There are two types of local-use unicast addresses:

1. Link-local addresses are used between on-link neighbors and for Neighbor Discovery (ND) processes.

2. Site-local addresses are used between nodes communicating with other nodes in the same site.

Link-local addresses are used by nodes when communicating with neighboring nodes on the same link. For example, on a single link IPv6 network with no router, link-local addresses are used to communicate between hosts on the link. The scope of a link-local address is the local link. Site-local addresses are equivalent to the IPv4 private address space. Site-local addresses are not reachable from other sites, and routers must not forward site-local traffic outside the site. Site-local addresses can be used in addition to global unicast addresses. The scope of a site-local address is the site.

3.1.2 Multicast Addresses

A multicast address identifies multiple interfaces. With the appropriate multicast routing topology, packets addressed to a multicast address are delivered to all interfaces

[6] Microsoft Windows Server 2003 White Paper, September 2003. Page 22.

that are identified by the address. A multicast address is used for one-to-many communication, with delivery to multiple interfaces. Interfaces may belong to more than one multicast group.

In IPv6, multicast traffic operates in the same way that it does in IPv4. Arbitrarily located IPv6 nodes can listen for multicast traffic on an arbitrary IPv6 multicast address. IPv6 nodes can listen to multiple multicast addresses at the same time. Nodes can join or leave a multicast group at any time.

3.1.3 Anycast Addresses

An anycast address also identifies multiple interfaces. With the appropriate routing topology, packets addressed to an anycast address are delivered to a single interface, the nearest interface that is identified by the address. The "nearest" interface is defined as being closest in terms of routing distance. An anycast address is used for one-to-"one-of-many" communication, with delivery to a single interface.

Packets addressed to an anycast address are forwarded by the routing infrastructure to the nearest interface to which the anycast address is assigned. In order to facilitate delivery, the routing infrastructure must be aware of the interfaces assigned anycast addresses and their "distance" in terms of routing metrics. At present, anycast addresses are only used as destination addresses and are only assigned to routers. Anycast addresses are assigned out of the unicast address space and the scope of an anycast address is the scope of the type of unicast address from which the anycast address is assigned.

3.2 Address Resolution

The Domain Name System (DNS) is used to resolve domain names (e.g., www.usmc.mil) by identifying the numeric address associated with it. Network routing is then possible based on the IP address returned by the DNS. DNS service is specific to IPv4 or IPv6. To resolve addresses across protocol boundaries, either a separate DNS service must exist for each Internet protocol or an application layer gateway must perform translation on addresses returned by the DNS answer message. DNS servers listing data for IPv4 (A records) and IPv6 (AAAA records) addresses can be configured to return IPv4, IPv6, or both addresses. The selection of which address type to return, or in which order, affects the type of IP traffic generated.

3.3 Features of IPv6

Systems currently operating on IPv4 were engineered to take advantage of features available in IPv4. Simply making these systems IPv6 capable will not take advantage of new features made possible by IPv6; applications ported to IPv6 will offer only the capabilities they had with IPv4. Planning and engineering efforts should provision for the enhanced feature set of IPv6. The following are features of the IPv6 protocol:[7]

3.3.1 New Header Format

[7] Microsoft Windows Server 2003 White Paper, September 2003. Pages 2-3.

The IPv6 header has a new format that is designed to keep header overhead to a minimum. This is achieved by moving both non-essential fields and optional fields to extension headers that are placed after the IPv6 header. The streamlined IPv6 header is more efficiently processed at intermediate routers.

3.3.2 Large Address Space

IPv6 has 128-bit (16-byte) source and destination IP addresses. Although 128 bits can express over 3.4×10^{38} possible combinations, the large address space of IPv6 has been designed to allow for multiple levels of subnetting and address allocation from the Internet backbone to the individual subnets within an organization. Even though only a small number of the possible addresses are currently allocated for use by hosts, there are plenty of addresses available for future use. With a much larger number of available addresses, address-conservation techniques, such as the deployment of NATs, are no longer necessary.

3.3.3 Efficient and Hierarchical Addressing and Routing Infrastructure

IPv6 global addresses used on the IPv6 portion of the Internet are designed to create an efficient, hierarchical, and summarizable routing infrastructure that is based on the common occurrence of multiple levels of Internet service providers.

3.3.4 Stateless and Stateful Address Configuration

To simplify host configuration, IPv6 supports both stateful address configuration, such as address configuration in the presence of a DHCP server, and stateless address configuration (address configuration in the absence of a DHCP server). With stateless address configuration, hosts on a link automatically configure themselves with IPv6 addresses for the link (called link-local addresses) and with addresses derived from prefixes advertised by local routers. Even in the absence of a router, hosts on the same link can automatically configure themselves with link-local addresses and communicate without manual configuration.

One of the most useful aspects of IPv6 is its ability to automatically configure itself, even without the use of a stateful configuration protocol such as Dynamic Host Configuration Protocol for IPv6 (DHCPv6). By default, an IPv6 host can configure a link-local address for each interface. By using router discovery, a host can also determine the addresses of routers, other configuration parameters, additional addresses, and on-link prefixes. Included in the Router Advertisement message is an indication of whether a stateful address configuration protocol should be used.

Address autoconfiguration can only be performed on multicast-capable interfaces.

3.3.5 Built-in Security

Support for IPSec is an IPv6 protocol suite requirement. This requirement provides a standards-based solution for network security needs and promotes interoperability between different IPv6 implementations.

3.3.6 Better Support for QoS

New fields in the IPv6 header define how traffic is handled and identified. Traffic identification using a Flow Label field in the IPv6 header allows routers to identify and provide special handling for packets belonging to a flow, a series of packets between a source and destination. Because the traffic is identified in the IPv6 header, support for QoS can be achieved even when the packet payload is encrypted through IPSec.

3.3.7 New Protocol for Neighboring Node Interaction

The Neighbor Discovery protocol for IPv6 is a series of Internet Control Message Protocol for IPv6 (ICMPv6) messages that manage the interaction of neighboring nodes (nodes on the same link). Neighbor Discovery replaces the broadcast-based Address Resolution Protocol (ARP), ICMPv4 Router Discovery, and ICMPv4 Redirect messages with efficient multicast and unicast Neighbor Discovery messages.

3.3.8 Extensibility

IPv6 can easily be extended for new features by adding extension headers after the IPv6 header. Unlike options in the IPv4 header, which can only support 40 bytes of options, the size of IPv6 extension headers is only constrained by the size of the IPv6 packet.

4 TRANSITION MECHANISMS FOR IPV6

IPv6 transition mechanisms are intended to provide a number of features, including:

- Incremental upgrade and deployment. Individual IPv4 hosts and routers may be upgraded to IPv6 one at a time without requiring any other hosts or routers to be upgraded at the same time. New IPv6 hosts and routers can be installed one by one.
- Minimal upgrade dependencies. The only prerequisite to upgrading hosts to IPv6 is that the DNS server must first be upgraded to handle IPv6 address records. There are no pre-requisites to upgrading routers.
- Easy Addressing. When existing installed IPv4 hosts or routers are upgraded to IPv6, they may continue to use their existing address.[8]

IPv6 provides an addressing structure called Compatibility Addresses that embeds IPv4 addresses within IPv6 addresses and encodes other information used by the transition mechanisms. The three methods available to enable IPv6 on an existing IPv4 network are Dual Stacking, Tunneling, and Translation. A combination of all three transition mechanisms will be used on Marine Corps networks.

4.1 Compatibility Addresses

To aid in the migration from IPv4 to IPv6 and the coexistence of both types of hosts, the following addresses are defined for IPv6:

4.1.1 IPv4-Compatible Address

The IPv4-compatible address, 0:0:0:0:0:0:*w.x.y.z* or ::*w.x.y.z* (where *w.x.y.z* is the dotted decimal representation of an IPv4 address), is used by IPv6/IPv4 nodes that are communicating using IPv6. IPv6/IPv4 nodes are nodes with both IPv4 and IPv6 protocols. When the IPv4-compatible address is used as an IPv6 destination, the IPv6 traffic is automatically encapsulated with an IPv4 header and sent to the destination using the IPv4 infrastructure.

4.1.2 IPv4-Mapped Address

The IPv4-mapped address, 0:0:0:0:0:FFFF:*w.x.y.z* or ::FFFF:*w.x.y.z*, is used to represent an IPv4-only node to an IPv6 node. It is used only for internal representation. The IPv4-mapped address is never used as a source or destination address of an IPv6 packet.

4.1.3 6to4 Address

The 6to4 address is used for communicating between two nodes running both IPv4 and IPv6 over an IPv4 routing infrastructure. The 6to4 address is formed by

[8] IP Next Generation Overview, accessible at http://playground.sun.com/pub/ipng/html/INET-IPng-Paper.html, by Robert M. Hinden, May 1995.

combining the prefix 2002::/16 with the 32 bits of a public IPv4 address of the node, forming a 48-bit prefix. 6to4 is a tunneling technique described in RFC 3056.

4.2 Dual Stacking IPv4 and IPv6

As stated in section 1.2, a host or router must use an implementation of both IPv4 and IPv6 in order to recognize and process both header formats. Dual Stacking is a model of deployment where all hosts and routers upgraded to IPv6 are "dual" capable. Dual Stacked hosts implement complete IPv4 and IPv6 protocol stacks. Due to the proliferation of IPv4 products, the transition to IPv6 will involve extensive enabling of dual-stack capability on networks. IPv6 provides support for the coexistence of both addressing schemes through Compatibility Addresses.

Since IP does not truly support two header formats for one packet, Dual Stacking actually implies the capability of a router or host to choose IPv4 or IPv6 for outbound traffic while retaining the ability to receive either header format.

4.3 Tunneling

Tunneling involves encapsulating IPv6 packets within IPv4 headers to carry them over segments of the end-to-end path where the routers have not yet been upgraded to IPv6. Tunneling allows isolated IPv6 hosts (i.e., located on a physical link which has no directly connected IPv6 router) to become fully functional IPv6 hosts by using an IPv4 multicast domain as their virtual local link to distant IPv6 networks.

4.4 Translation

Translation occurs at application layer gateways between portions of the network that are using different Internet protocols and allows the deployment of hosts that support only IPv6 on IPv4 networks. Because some features available in IPv6 do not translate to IPv4, Translation has limited value. IPv6 features such as flow control, multicast and unicast neighbor discovery, IPSec, and extension headers may not be supported when translated into an IPv4 network.

5 POTENTIAL CONSTRAINTS TO IMPLEMENTING IPV6

5.1 Network Appliance Memory

While support for IPv4 and IPv6 on the same router is generally a matter of upgrading the IOS and enabling routing features, there is a hardware requirement as well. Memory in routers is used to process routing requests, manage queues, and route traffic. Dual Stacking routers to support routing tables for IPv4 and IPv6 will require additional memory in most cases. The cost associated with adding memory to routers, servers, and other network appliances should not be overlooked when planning IPv6 transition.

5.2 Operating Systems

Production support for IPv6 does not exist in most operating systems in use in the Marine Corps today. Windows XP with service pack 1 or later is the first release of Microsoft operating system that advertises production IPv6 capability. Microsoft does not plan to support IPv6 with earlier versions of operating systems, including Windows 2000. Interoperability issues are likely to arise between early releases of IPv6 capable operating systems developed by different vendors.

5.3 Application Porting and Adding IPv6 Capability[9]

Applications can be developed so they are "agnostic" to the IP version in use. Software should rely on the IP stacks in COTS operating systems to the extent possible.

Many COTS and open source software applications already have some level of IPv6 support built in. The development cycles for this class of software tend to be relatively rapid so new versions should be continually examined for IPv6 functionality and maturity.

Few GOTS applications have IPv6 capabilities at this time. Porting will not enable all the functionality and capabilities of IPv6 in GOTS applications, just those that are already available with the current IPv4 software. Enabling advanced IPv6 features such as mobility, anycast addressing, and Quality of Service (QoS) will likely require additional software development, testing, and certification.

Adding IPv6 capability to applications while retaining support for IPv4 may not require significant additional effort. Vendors have stated the effort was "more tedious than difficult". Developers can begin by downloading a scanning tool such as Sun's IPv6 Socket Scrubber, Microsoft's Checkv4, and Compaq's IPv6 Porting Assistant for Tru64 Unix. These tools operate on source code to identify areas needing modification to support IPv6. Modifications can then be made to the source code and the software recompiled, tested, and certified for use. The Microsoft publication IPv6 Guide for

[9] Much of the content for this section was extracted from "Projected Impacts of IPv6 on the USN and USMC Enterprise" by Michael P. Brig. SPAWAR Systems Center Charleston.

Windows Sockets Applications[10] divides the software porting effort into five areas. These areas are:

1. Changing data structures.
2. Function calls.
3. Use of hard-coded IPv4 addresses.
4. User interface issues.
5. Elimination of user interface issues.

Scanning tools can identify lines of IPv4 socket code in applications and review and modify code with the new IPv6 socket API. Porting a socket application typically only requires a few lines of code change. Scanning tools can test code to help find any IPv4 dependencies and modify code so the application will be able to use IPv6.

It should be noted Microsoft has no plans to provide the IPv6 software libraries and function calls for Windows 2000. This complements Microsoft's strategy of providing a "production" IPv6 stack only for Windows XP (with Service Pack 1 or later) and later versions of the Windows OS.

Some GOTS applications are actually conglomerations of COTS applications and government developed code. Porting such an application might require a considerable coordination effort and involve multiple interdependencies. It is impractical to port a GOTS application if a critical COTS software component was not yet ported to IPv6. In this case, a Program Manager (PM) might simply replace that software component with a different commercial alternative but then other code changes would likely be required.

5.4 Compatibility of IPv4 and IPv6

Upper layer protocol checksums must support IPv6 headers. The current implementation of TCP and UDP for IPv4 incorporates into their checksum calculation a pseudo-header that includes both the IPv4 Source Address and Destination Address fields. This checksum calculation must be modified for TCP and UDP traffic sent over IPv6 to include IPv6 addresses.[11] In most cases interface with these upper layer protocols is provided by the operating system. Applications with hard coded upper layer protocol implementations will require development, testing and certification to enable IPv6 support.

Translation from IPv6 to IPv4 can result in loss of transfer of some IPv6 capabilities and interfere with end-to-end application performance. Refer to section 4.4 for additional considerations necessitated by network translation.

5.5 Security

Currently employed network security devices such as the Network Encryption System (NES), Fastlane, Taclane, and KY trunk encryptors must support IPv6. Firewalls must be upgraded or changed to support IPv6. Tunneling may present security and

[10] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winsock/winsock/ipv6_guide_for_windows_sockets_applications_2.asp

[11] Microsoft Windows Server 2003 White Paper, September 2003. Page 34.

implementation challenges as encapsulated packets pass through firewalls. Firewalls will take time to examine encapsulated packets, increasing network latency and potentially affecting application performance.

Adoption of IPv6 on Marine Corps networks opens vulnerabilities from external IPv6 networks. Procedures for protecting from attacks and probes from IPv6 sources must be incorporated into current Information Assurance Vulnerability Alert (IAVA) and Naval Incidence Response Team (NAVCIRT) advisory procedures.

Dual stacking networks introduces vulnerabilities that would not exist on a single protocol network. Gateways providing protocol translation introduce another layer of complexity in the architecture and, therefore, another point that could be exploited by hackers or malicious code.

5.6 Technical and Programmatic Risks

Any IPv6 transition event that may delay scheduled program events introduces programmatic risk. Technical risks result if required capabilities are not available within an allotted timeframe. Examples of IPv6 transition events that may impact program schedules are:

- Availability of IPv6 capable hardware or software
- Software that must be recoded to support IPv6
- Operating System support for IPv6
- Reliance on an externally managed system that must transition first
- Proprietary implementations of IP that must be reengineered to support IPv6
- Security risks introduced by transition to IPv6

5.7 Implications of IPv6 Transition to Programs and Applications

Appendix B provides a checklist for identifying the IPv6 transition efforts needed for programs of record. Program Managers will use the checklist to focus engineers and planners on actions required to make their programs IPv6 capable. Enterprise planners will aggregate completed checklists to identify dependencies that will affect transition timelines. Completed templates will also facilitate efficient use of transition resources; both technical expertise and funding will be limited. Appendix C contains a similar checklist for software applications.